Report
Song Recommendation Model

Atharva Tendle & Ayush Manish Agrawal

We wanted to build a recommendation system which would be able to classify user's musical taste based on a song's acoustic features. We were aiming to build a model that is similar to spotify's discover weekly, we wanted to augment it further by suggesting other random songs/genres for a user to explore.

Our problem is both classification and regression. We first classify the songs based on the features and then use regression to predict similar songs that user might like based on previous interests.

The reason why this problem is important might seem silly at first to serious researchers but to us it was something quintessential. Music is something which has been constant through mankind's history and having a efficient way to listen to music which matches one's tastes is a very important problem.

This problem interests us because we have noticed that sometimes the spotify's discover weekly playlist suggests bad songs for our tastes and does not suggest exploratory songs which we might like and we thought that it was important to make it better. We decided to take our first step using this project as the base to understanding what is possible and what can we do to improve the song recommendation.

Dataset:

- **Dataset:** FMA - Data Set for music analysis
- **Dataset Source:** https://arxiv.org/abs/1612.01840
- **Features:** 8 Features
- **Examples:** ~13000
- **Feature Type:** Real Data (Song Features like acousticness, dancebility etc.)
- **Feature Recoding:** We did not recode any features
- **Labels:** [1, 0]: 1 = User Likes the song, 0 = User Dislikes the song. We created these labels via genre based randomization which incorporated a biased randomization for like and dislike values.

Methods:

- We implemented k-Nearest Neighbors, Logistic Regression and Multilayer Perceptron Classifier.
- We implemented cross-validation to gauge our model and also for hyperparameter searching.
- We tried to investigate if we needed to normalize and/or scale the data but after learning about the models and examining our data we came to the conclusion that we did not need to normalize and/or scale our data. We did not transform any features.

Results:

- We computed the Accuracy score and the confusion matrix and we found that our accuracy ranges between 60-70%.

- The low accuracy was mainly due to the dataset issues explained below.

- Our hyperparameters performed decently considering the various dataset related issues we faced and helped in achieving a decent accuracy.

- We generally had strong True Positive values in our Confusion Matrix which was good thing.

Issues and Insights:

- We ran into a lot of problems when it came to our dataset. Since the Million Song Dataset was really complex to parse, we had to settle for a different smaller dataset. Other than being smaller dataset, there was a high bias for genres like rock, pop and hip-hop. This bias caused our dataset to not understand and learn about the features for other genres which resulted in a poor accuracy during prediction.

- We also had difficulty in implementing the MLP due to fairly insufficient knowledge. This led to a lot of time being spent on understanding the inner workings of how to model the MLPClassifier in the most optimum manner for our project.

- The best way to improve on these issues would have been figuring out a way to parse the Million Songs Dataset and use that to run our Project.

- We wish to parse the MSD and then run that on our project and see how it performs, then we would probably improve parts of our project to reach our desired goal.